Theoretical Computer Science 116 (1993) 373–398
Elsevier

Affine automata and related techniques for generation of complex images

Karel Culik II and Simant Dube

Department of Computer Science, University of South Carolina, Columbia, SC 29208, USA

Communicated by M. Nivat
Received July 1990
Revised July 1991

Abstract

Culik II, K. and S. Dube, Affine automata and related techniques for generation of complex images, Theoretical Computer Science 116 (1993) 373–398.

In this paper, we introduce probabilistic affine automata (PAA), which are probabilistic finite generators having transitions labeled with affine transformations. It is shown that PAA are capable of generating highly complex images. Barnsley's (1988) IFS method to generate fractals is a special case of PAA when the automaton happens to have only a single state. A number of theoretical results on PAA are shown. The relationship between PAA, mutually recursive function systems (MRFS) and affine regular sets is investigated.

1. Introduction

Recently, "fractal geometry", introduced by Mandelbrot [15], has been getting increased attention in relation to the study of deterministic chaos (complex systems) [13]. The relation of fractal geometry to classical geometry is similar to the relation of classical physics, which handles primarily phenomena described by linear differential equations, to the new "chaos" physics. Chaos physics studies complex phenomena, mathematically described by nonlinear differential equations, like the flow of gases. Classical geometry is good at handling "man-made" objects like polygons, circles, etc. The new fractal geometry should handle well all the classical objects as well as those of fractal (recurrent) type. Examples are H-trees, Sierpinski triangles and also all natural

Correspondence to: K. Culik II, Department of Computer Science, University of South Carolina, Columbia, SC 29208, USA.

objects like plants, trees, clouds, mountains, etc. The study of fractal geometry was pioneered by Mandelbrot [15] and the study of practical "computational fractal geometry" by Barnsley [1]. The latter introduced the iterative function systems (IFS) that are used to define an object (an image) as the limit (attractor) of a "chaotic process". He has used IFS to generate exclusively deterministic fractals. Voss et al. [3] have considered techniques to generate random fractals. Barnsley's [1] hyperbolic IFS is specified by several affine transformations, and the attractor is the limit of the sequence generated from an arbitrary starting point by randomly choosing and applying these affine transformations.

The encoding of images by IFS and other methods has potential for practical applications because it allows compression of data and their efficient processing. For example, from an IFS description of an image it is possible to regenerate effectively not only the original image but also its various modifications, e.g. a view from a different angle. Applications of this are being developed not only to computer graphics [4] but also to the compression of videos, to medical imaging, and to high-resolution TV, etc.

The aim of this paper is to introduce some powerful generalizations of the IFS method, which define a much bigger class of interesting images.

The paper is organized as follows. We first discuss the formal notions of an image in Section 2. A black and white image is formalized as a compact set and a texture (color) image is formalized as a normalized measure (greyness density). We then introduce Barnsley's IFS method to generate fractals.

In Section 3, we introduce probabilistic affine automaton (PAA), which is informally a probabilistic finite generator whose input symbols are affine transformations. PAA are a generalization of recurrent IFS introduced in [2] and, for many images, give a much more concise description. We will show a number of results about PAA. Some of them show that PAA is a robust notion, since various modifications of the PAA generate the same family of compact sets. This is true even for MRFS discussed later.

We consider two more generalizations of IFS. In Section 4, we discuss briefly a generalization of Barnsley's IFS called affine regular sets, which define a bigger class of images of more complex geometries. Intuitively, an affine regular set generates an image based on a finite set of affine transformations that are applied in an order controlled by a regular set. The other one, as introduced in Section 5, is called a mutually recursive function system (MRFS) and is given by a number of "variables" which are defined in terms of each other as unions under affine transformations. MRFS on arbitrary complete metric spaces have been considered in [16], where some results on Hausdorff dimension of objects defined by MRFS are shown. We consider both deterministic and probabilistic variations of MRFS.

As expected, PAA and affine regular sets have the same descriptive power; however, surprisingly all these generalizations – PAA, affine regular sets and MRFS, turn out to be exactly equivalent in terms of their power to generate image (as compact sets), as shown in Sections 5 and 6.

Barnsley's [1] collage theorem gives the mathematical basis for inferring a concise IFS-description of any given image, which includes texture or color. The collage theorem can be extended also to affine automata and mutually recursive IFS; however, we presently do not have any efficient method of encoding arbitrary images by these methods.

In Section 7 we give some applications of PAA and PMRFS. This includes image generation and compression, as illustrated by two examples. One can generate some interesting "texture" images, where texture is a quality describing the "graininess" of an image, by using the hybrid algorithm, which combines both probabilistic and deterministic MRFS and can, furthermore, use more than one MRFS. Rational expression, as introduced in [9] to define images, are a special case of affine expressions (affine regular sets). This leads to an efficient implementation of rational expressions by PAA or PMRFS. This implementation does not use the bit-by-bit approach and, hence, yields algorithms that, using standard (numerically oriented) software and hardware, are almost as fast as Barnsley's. Finally, we mention that, under certain conditions [8], MRFS can simulate another technique for generation of images, which is based on L-systems (string rewriting systems) [11, 17].

2. Preliminaries

2.1. Two notions of an image

Following [1], we introduce two different formalizations of an image:

(1) Given a compact metric space (X, d), an image is a compact subset of X. The quality of an approximation for such images is measured by the Hausdorff metric $h(d)$ on the compact metric space $\mathscr{H}(X)$ of the nonempty compact subsets of X. This is a formalization of such an image as consisting of black and white regions. A finite approximation of such an image on the computer screen is an assignment of 0 (white) or 1 (black) to each pixel of a matrix of pixels.

(2) Given a compact metric space (X, d), an image is a normalized invariant measure on X, that is, an additive function f defined on the Borel subsets of X such that $f(X)=1$ (see [1] for more details). The quality of an approximation for such images is measured by the Hutchinson metric d_{H} on the compact metric space of all the normalized measures on X. This is a formalization of an image as a texture, either of the various tones of grey or of colors. A finite approximation of such an image on the computer screen is an assignment of grey tones (or colors) to each pixel. Here each pixel represents a small subsquare of the space X, the measure of which is translated into the grey level (color) assigned to the pixel.

2.2. Iterative function systems

A space X together with a real-valued function $d: X \times X \to \mathbf{R}$, which measures the *distance* between pairs of points x and y in X, is called a *metric space*. In this paper, we will be concerned with metric spaces ($\mathbf{R}^n$, Euclidean), where $n \geqslant 1$ and $\mathbf{R}$ is the set of real numbers.

A 2-dimensional (2-D) *affine transformation* $w: \mathbf{R}^2 \to \mathbf{R}^2$ is defined by

$$w\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_{11}x + a_{12}y + b_1 \\ a_{21}x + a_{22}y + b_2 \end{bmatrix},$$

where a_{ij}'s and b_i's are real constants [1]. Similarly, a 1-dimensional (1-D) affine transformation $w: \mathbf{R} \to \mathbf{R}$ is defined by $w(x) = ax + b$, where a and b are real constants.

A transformation $f: X \to X$ on a metric space (X, d) is called a *contractive mapping* if there is a constant $0 \leqslant s < 1$ such that

$$d(f(x), f(y)) \leqslant s \cdot d(x, y) \quad \text{for all } x, y \in X.$$

Any such number s is called a *contractivity factor* for f.

Let (X, d) be a compact metric space. Then $\mathscr{H}(X)$ denotes the space whose points are the compact nonempty subsets of X. Let $x \in X$ and $B \in \mathscr{H}(X)$. The distance from the point x to the set B is defined to be

$$d(x, B) = \min\{d(x, y): y \in B\}.$$

Now, let $A \in \mathscr{H}(X)$. The distance from the set A to the set B is defined to be

$$d(A, B) = \max\{d(x, B): x \in A\}.$$

The *Hausdorff distance* between the sets A and B is defined by

$$h(A, B) = \max\{d(A, B), d(B, A)\}.$$

It can be shown that $(\mathscr{H}(X), h(d))$ is a compact metric space [1].

A (hyperbolic) *iterated function system* (IFS) consists of a compact metric space (X, d), together with a finite set of contractive mappings $w_n: X \to X$, with respective contractive factors s_n, for $n = 1, 2, \ldots, N$. The notation for the IFS defined is $\{X; w_n, n = 1, 2, \ldots, N\}$ and its contractivity factor is $s = \max\{s_n: n = 1, 2, \ldots, N\}$. We will also consider probabilistic IFS, in which a probability $p_i > 0$ is associated with each mapping w_i such that $\sum_1^N p_i = 1$.

In our examples we will use the metric spaces ([0, 1], Euclidean) and ($[0, 1]^2$, Euclidean), for 1-D and 2-D images, respectively, and affine transformations. The transformation $W: \mathscr{H}(X) \to \mathscr{H}(X)$, defined by

$$W(B) = \bigcup_{n=1}^{N} w_n(B),$$

for all $B \in \mathscr{H}(X)$, is a contraction mapping on the compact metric space $(\mathscr{H}(X), h(d))$, where $h(d)$ is the Hausdorff distance. Its unique fixed point, $A \in \mathscr{H}(X)$, is called the *attractor* of the IFS and is the geometric object defined by the IFS. This geometric object could be self-similar and, hence, a fractal may be defined by an IFS [1].

This basic mathematical characterization of a fractal to be the attractor of an IFS provides two algorithms to generate fractals [1]. Let $\{X; w_1, w_2, \ldots, w_N\}$ be an IFS with probability p_i associated with w_i for $i=0, 1, \ldots, N$.

The *deterministic* algorithm starts by choosing a compact set $A_0 \subset X$. Then one computes successively $A_n = W^n(A_0)$ according to

$$A_{n+1} = \bigcup_{j=1}^{N} w_j(A_n) \quad \text{for } n=1, 2, \ldots$$

The sequence $\{A_n\}$ converges to the attractor of the IFS in the Hausdorff metric. Note that the probabilities assigned to the mappings play no role in the deterministic algorithm.

The random iteration (or chaos game) algorithm starts by choosing a point $x_0 \in X$. Then one chooses, recursively and independently,

$$x_n \in \{w_1(x_{n-1}), w_2(x_{n-1}), \ldots, w_N(x_{n-1})\} \quad \text{for } n=1, 2, 3, \ldots,$$

where the probability of the event $x_n = w_i(x_{n-1})$ is p_i. The collection of points $\{x_n\}_{n=0}^{\infty}$ converges to the attractor of the IFS. This algorithm also defines the texture of an image as an invariant measure on Borel subsets of X [1].

Example. The Sierpinski triangle, a subset of $[0, 1]^2$, is the attractor of the IFS specified by the affine transformations $w_1(x, y) = (0.5x, 0.5y)$, $w_2(x, y) = (0.5x + 0.5, 0.5y)$ and $w_3(x, y) = (0.5x, 0.5y + 0.5)$. The output of the deterministic algorithm after 8 iterations is shown in Fig. 1(a).

Example. Fig. 1(b) shows a self-similar fern, generated by 80 000 iterations of the chaos game algorithm on the IFS specified by the four affine transformations

$$w_1(x, y) = (0, 0.16y),$$

$$w_2(x, y) = (0.85x + 0.04y, -0.04x + 0.85y + 0.16),$$

$$w_3(x, y) = (-0.15x + 0.28y, 0.26x + 0.24y + 0.044),$$

$$w_4(x, y) = (0.2x - 0.26y, 0.23x + 0.22y + 0.16).$$

The corresponding probabilities are 0.01, 0.85, 0.07 and 0.07, respectively.

In [4, 12] IFS is generalized to *recurrent* IFS. In a recurrent (or Markov) IFS, we are given a set of N contractive mappings, w_i's, along with a $N \times N$ row-stochastic matrix denoted by (p_{ij}). The value p_{ij} gives the probability of applying w_j when in the last iterative step w_i was applied.

2.3. *Languages of infinite words*

We assume that the reader is familiar with basic formal language theory, in which languages are defined as sets of *finite* words. This computation domain can be

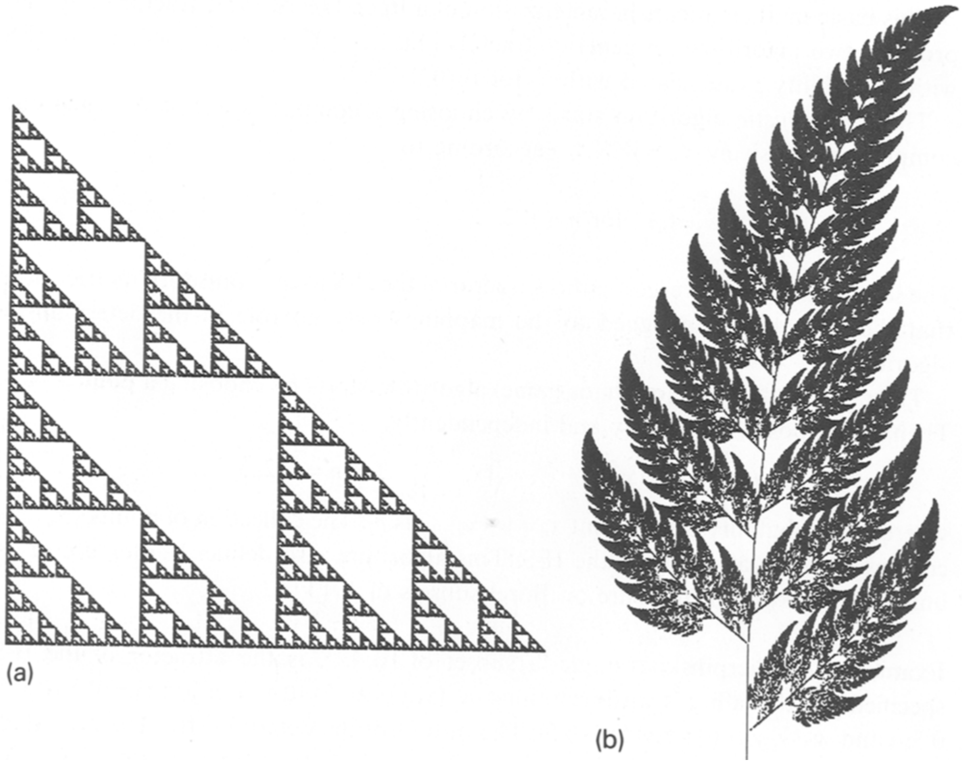


Fig. 1. Examples of images generated by IFS.

extended by adding the set of infinite strings Σ^ω [7, 10]. Formally, Σ^ω denotes all infinite (ω-length) strings $\sigma = \prod_{i=1}^{\infty} a_i$, $a_i \in \Sigma$, over Σ. An element σ of Σ^ω is called an *ω-word* or *ω-string*. An *ω-language* is any subset of Σ^ω. The set of both finite and infinite strings is denoted by $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. The superscript ω means infinite repetition, e.g. $(00)^*1^\omega$ denotes an ω-set of strings which have an even number of zeroes followed by an infinite number of consecutive ones. Later we will interpret finite strings as rational numbers and ω-strings as real numbers.

For any language $L \subseteq \Sigma^*$, define

$$L^\omega = \left\{ \sigma \in \Sigma^\omega \;\middle|\; \sigma = \prod_{i=1}^{\infty} x_i,\ \forall i,\ x_i \in L \right\}.$$

Therefore, L^ω consists of all ω-strings obtained by concatenating words from L in an infinite sequence.

For any family $\mathscr{L}$ of languages over alphabet Σ, the ω-Kleene closure of $\mathscr{L}$, denoted by ω-KC($\mathscr{L}$), is given by

$$\omega\text{-KC}(\mathscr{L}) = \left\{ L \subseteq \Sigma^{\omega} \,\middle|\, L = \bigcup_{i=1}^{k} V_i W_i^{\omega} \text{ for some } V_i, W_i \in \mathscr{L}, \quad i = 1, 2, \ldots, k;\ k = 1, 2, \ldots \right\}.$$

If $\mathscr{L}$ is the family of regular languages, then ω-KC($\mathscr{L}$) is called the family of *ω-regular languages* [7]. In a straightforward generalization, we may define *ω-rational expressions* for *ω-rational relations.*

The ω-regular languages are exactly the languages accepted by the ω-finite automata, which are defined as follows.

An *ω-finite automaton* (ω-FA) is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where Q is the finite set of states, Σ is the input alphabet, δ is a mapping from $Q \times \Sigma$ to 2^Q, q_0 is the initial state and F is the set of final states.

An ω-word is accepted by an ω-FA if on reading the input ω-word the ω-FA enters a final state infinitely many times.

An ω-word τ is called an *adherence* of a language L if τ has infinitely many prefixes such that each of these prefixes is also a prefix of some word in L. Formally, define $\text{Prefix}(y) = \{x \in \Sigma^* \mid x \text{ is a prefix of } y\}$. Note that y can be both a finite or an infinite word. An ω-word τ is an adherence of L iff $\text{Prefix}(\tau) \subseteq \text{Prefix}(L)$. The set of all adherences of L, denoted by adherence(L), is called the adherence set of L. The adherence set of any regular language is ω-regular [6, 10]. Note that the adherence set of a language L, which is accepted by an FA M, is accepted by an ω-FA M', which is obtained from M by discarding all those states which do not have an outgoing transition and by making every state a final state.

3. Probabilistic affine automata

In Barnsley's chaos game algorithm, in each iteration, the next affine transformation to be applied on the last generated point is chosen from the same fixed set of affine transformations. Moreover, the probability of choosing a particular transformation is always the same in each iteration. A significant improvement can be made on this algorithm, in terms of its capacity to generate images, by changing this set of transformations and the associated probabilities, on each iterative step.

One can, therefore, think of employing a finite automaton, whose finite control determines the set of transformations and the associated probabilities for each iterative step. Furthermore, one may mark a subset of the states of the automaton as "final" or "display" states, and display only the points generated at these states. One would expect this to give additional power to generate images, but in this paper it is shown later that this division of states into final and nonfinal states is really not essential. The above idea to generalize the chaos game algorithm is formalized in the definition of probabilistic affine automata.

Let G be a directed graph, possibly representing the transition diagram of a finite automaton. Let the edges of G be labeled with affine transformations. If a path P in G, of length $n \geq 1$, is labeled with the sequence of affine transformations $w_{i_1}, w_{i_2}, \ldots, w_{i_n}$ of respective contractivity factors $s_{i_1}, s_{i_2}, \ldots, s_{i_n}$, then the path P is said to have a contractivity factor of $s = s_{i_1} \times s_{i_2} \times \cdots \times s_{i_n}$. The digraph G is said to satisfy *loop contractivity* condition if every loop in G has a contractivity strictly less than one.

Definition 3.1. A *probabilistic affine automaton* (PAA) is a 6-tuple $M = (X, S, \Sigma, \delta, P, F)$, where X is a compact metric space, $Q = \{q_1, q_2, \ldots, q_n\}$ is the set of *states*, Σ is a finite set of affine transformations $w_i : X \to X$, for $i = 1, 2, \ldots, m$, $\delta : Q \times \Sigma \to Q$ is the *state transition function* such that the underlying transition diagram satisfies the conditions of loop contractivity and strong connectivity, P is an $n \times m \times n$ stochastic matrix such that, for each i, $\sum_{k=1}^{n} \sum_{j=1}^{m} P(i, j, k) = 1$, and $F \subseteq Q$ is the set of final states.

The value $P(i, j, k)$ is the probability of the transition from the state q_i to the state q_k by transformation w_j. Note that the probabilities of outgoing transitions for each state sum to unity. In other words, a PAA is a probabilistic finite generator whose input alphabet is a set of affine transformations and which satisfies the conditions of strong connectivity and loop contractivity.

A PAA M generates an image on the basis of the following algorithm, which is a generalization of the chaos game algorithm. A point in X is randomly chosen; call it x_0. Any state of M can be randomly chosen to be the initial state. The PAA then generates a *sequence* of points just like in chaos game algorithm, the only difference being that, at any step, one of the outgoing transitions of the current state of the PAA is probabilistically chosen according to the associated probabilities, and the affine transformation labeling the chosen transition is applied to the last generated point. The finite control of M then changes its current state and the point is displayed if this new current state is a final state. In other words, if q is the current state and x is the last generated point and if the transition $\delta(q, w) = p$ is chosen, then $w(x)$ is the next point generated, and it is considered to be in the image (or, to be precise, in an approximation to the image) defined by M if p is a final state.

This process yields a limiting sequence of points, say $S_0 = \{x_0, x_1, x_2, \ldots\}$ generated at final states. This set of points so generated is an approximation to the attractor of the given PAA, denoted by $A(M)$.

To be mathematically precise, a point is in $A(M)$ iff its every neighborhood is visited infinitely many times with probability almost one during the execution of the algorithm.

Therefore, if $B(x, \varepsilon)$ denotes the closed ball of radius ε with center at x,

$$B(x, \varepsilon) = \{ y \in X \mid d(x, y) \leq \varepsilon \},$$

then,

$$A(M)=\{x\in X\mid \forall\varepsilon>0,\ \delta>0,\ \text{a point } y\in B(x,\varepsilon) \text{ is generated with probability } 1-\delta\}.$$

In [1], the texture of an image generated by an IFS is defined to be a measure on Borel subsets of X. Mathematically, the set of Borel subsets $\mathscr{B}$ is the σ-field generated from the open subsets of X by the operations of complementation and countable union.

A measure on Borel subsets is a real nonnegative function $\mu:\mathscr{B}\rightarrow[0,\infty)$ that is additive on a countable union of pairwise disjoint Borel sets. We extend this definition of the texture of images of PAA as follows. For details see [1]. Let B be a Borel subset of X. Let $\{x_0, x_1, x_2, \dots\}$ be a sequence of points generated by a PAA M. Define

$$\mathscr{N}(B,n)=\text{number of points in } \{x_0,x_1,\dots,x_n\}\cap B \quad \text{for } n=0,1,2,\dots$$

Then, with probability one,

$$\mu(B)=\lim_{n\rightarrow\infty}\{\mathscr{N}(B,n)/(n+1)\},$$

for all starting points x_0.

Informally, $\mu(B)$ is the "mass" of B, which is the proportion of iteration steps, when running the chaos game algorithm on M, which produce points in B. Points "fall" in different subsets of X according to the probabilities on the transitions; this notion is mathematically formalized in the definition of the texture of an image as a measure.

Therefore, the image defined by a PAA M can be viewed either as a compact set or as a measure. The result connecting these two definitions, which says that the support of the measure defined by an IFS is precisely the compact set defined by the same IFS (see [1]), also holds for PAA.

It is an easy observation that an IFS is a special case of a PAA.

Lemma 3.2. *Let $\{X;\ w_1, w_2, \dots, w_n\}$ be an IFS with probabilities p_i, $1\leqslant i\leqslant n$. Then there exists an equivalent PAA defining the same image (both as a compact set and a measure).*

Proof. The proof follows from the simple observation that an IFS can be implemented by a single-state PAA. The equivalent PAA has one state, which is the only initial as well as final state, with n self-loops as transitions, the ith transition being labeled with the transformation w_i and probability p_i, for $i=1,2,\dots,n$. This PAA is not only equivalent to the given IFS in terms of the equivalence of their attractors but also in terms of the texture of the image. □

Now we show that PAA are closed under invertible affine transformations.

Theorem 3.3. *The family of images (as compact sets) generated by PAA is closed under invertible affine transformations.*

Proof. Let $M=(X, Q, \Sigma, \delta, P, F)$, where $\Sigma=\{w_1, w_2, \ldots, w_n\}$, be a PAA and τ an invertible affine transformation. We construct a PAA $M'=(X, Q, \Sigma', \delta, P, F)$, where $\Sigma'=\{w'_1, w'_2, \ldots, w'_n\}$, with $w'_i=\tau \circ w_i \circ \tau^{-1}$. Now, let $x_0, x_1, x_2, \ldots$ be a sequence generated by M, and let $y_i=\tau(x_i)$ for $i \geqslant 0$. If $x_{i+1}=w_{j_i}(x_i)$, then $y_{i+1}=\tau(w_{j_i}(\tau^{-1}(y_i)))=\tau(x_{i+1})$ for all $i \geqslant 0$. Therefore, the sequence $y_0, y_1, y_2, \ldots$ can be generated by M' and, consequently, $A(M')=\tau(A(M))$. □

Theorem 3.3 is valid also for images as measures, assuming natural extension of transformations to measures.

One may expect that, by considering only those points which are generated at some selected states marked as "final", it is possible to define a bigger class of images (as compact sets) by PAA. However, it turns out that this division of states into final and nonfinal ones is not essential for a large class of PAA – namely, those PAA which have their transitions between two different states (i.e. not considering self-loops) labeled with invertible affine transformations. A (2-D) noninvertible affine transformation squeezes a parallelogram into a line or a point. Therefore, if there is a noninvertible transformation w from state p to state q, then there is some loss of "information" about the point generated at state p when w is applied to it. In practice, one seldom wants to have such a PAA; therefore, the restriction to PAA with only invertible transformations on transitions (except self-loops) is a minor one.

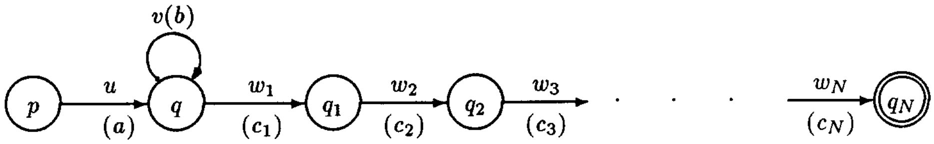
We do not know whether the following theorem generalizes to images considered as measures.

Theorem 3.4. *Let M be a PAA such that all its transitions, with the possible exception of its self-loops, are labeled with invertible affine transformations. Then there exists a PAA M' in which every state is a final state and which defines the same image as defined by M (as a compact set).*

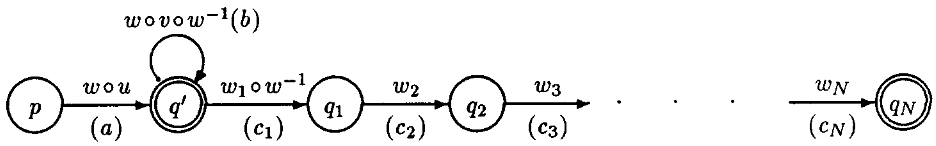
Proof. We will show how M can be converted into M' in a step-by-step manner such that in each step a nonfinal state is removed in such a way that loop contractivity and strong connectivity conditions are preserved and the resulting PAA defines the same image.

Pick any nonfinal state in M. Let it be q. We consider any simple path from q to a final state. In Fig. 2a, a part of the transition diagram of M containing q is shown, in which we are considering the simple path $\delta=qq_1q_2 \ldots q_N$ of length N. The corresponding probabilities are shown in parentheses.

Let $w=w_N \circ w_{N-1} \circ \cdots \circ w_2 \circ w_1$ be the composition of transformations labeling the path δ.



(a)



(b)

Fig. 2. Replacing a nonfinal state q by a final state q'.

A new final state q' is created which has its transitions defined in the following fashion:

(1) If q has an incoming transition, say from state p labeled with transformation u and having probability a, then q' also has an incoming transition from p labeled with transformation $w \circ u$ and having the same probability a.

(2) If q has a self-loop, say labeled with transformation v and having probability b, then q' also has a self-loop labeled with transformation $w \circ v \circ w^{-1}$ and having the same probability b.

(3) If q has an outgoing transition, say to state p, labeled with transformation u and having probability c, then q' has an outgoing transition to state p labeled with $u \circ w^{-1}$ and having the same probability c.

The new state q' has exactly the transitions defined by the above rules. The modified portion of the transition diagram of M is shown in Fig. 2b.

Similarly, we may consider each and every simple path from q to some final state and create a new final state for each path, although it suffices to consider only one such path. The state q is then considered to be deleted.

Now we claim that the resulting PAA preserves strong connectivity and loop contractivity. It is easy to see that the resulting PAA is still strongly connected. To verify that loop contractivity is preserved, note that if transformations x and y have contractivity factors s_1 and s_2, respectively, then $x \circ y$ has contractivity factor $s_1 \times s_2$.

Furthermore, the resulting PAA defines the same image as defined by the original PAA, as a compact set. To see this, note that a point is generated at a final state in Fig. 2a iff it is generated at a final state in Fig. 2b. This is because a path ends in a final state in Fig. 2a iff there is a path, with the same label, ending in a final state in Fig. 2b. In the modified PAA, although the points are generated earlier and more frequently.

The above algorithm is applied again and again to remove the remaining nonfinal states one by one. It terminates when there are no more nonfinal states, yielding the desired PAA M' which is equivalent to M. □

In [2], Markov chain or recurrent IFS are studied and, in [12], the Hausdorff–Besikovitch dimension of such generalized IFS is computed. Recurrent IFS have a strong connection with PAA. Recall that, in a recurrent IFS, we are given a set of N contractive affine transformations, w_i's, along with an $N \times N$ row-stochastic matrix P, where $P(i,j)$ gives the probability of applying w_j when, in the last iterative step, w_i was applied. All the points generated are considered to be an approximation to the attractor of the recurrent IFS and, therefore, there is no additional control of final states as in PAA.

We first note that the condition that every transformation is contractive causes a loss of generative power compared to the less restrictive condition of loop contractivity. However, the latter is still strong enough to ensure the existence of a unique attractor. This will be shown in more detail in the next section. Moreover, in order to show that a PAA, with contractive transformations, can be simulated by a recurrent IFS, we need to use recurrent IFS that allow several copies of the same transformation, i.e. whose transformations need not be unique.

Theorem 3.5. *For every PAA with only contractive transformations, there exists effectively a recurrent IFS that generates the same image (both as a compact set and a measure).*

Proof. Clearly, recurrent IFS as defined in [2] are a subset of PAA. Now, let us assume that we are allowed to specify a recurrent IFS by a set of affine transformations which need not necessarily be distinct. Under this assumption, for every PAA whose all transformations are contractive, we can construct an equivalent recurrent IFS as follows: Let M be a PAA with n transitions in its diagram. If an affine transformation u is the label of k different transitions, then we give k different names to the same transformation, i.e. we consider $u_1, u_2, \ldots, u_k$, which are just different "names" for the same affine transformation. Thus, each transition is labeled with a unique name and, in total, we have n names. Then we construct an $n \times n$ stochastic matrix P on these new transformations. If there is a state which has an incoming transition labeled with w_i and an outgoing transition labeled with w_j, then the value $P(i,j)$ is the probability assigned to the outgoing transition in the original PAA, else it is zero. Clearly, the resulting recurrent IFS is equivalent to original PAA. □

Note that the equivalent IFS is much bigger, as its underlying graph will have as many nodes as there are edges in the PAA. Therefore, we have a possible quadratic increase in the size of the description of the same image.

A transformation $w: X \to X$ is called a similitude of contractivity s if

$$d(w(x), w(y)) = s \cdot d(x, y) \quad \text{for all } x, y \in X.$$

If all the affine transformations of a PAA are similitudes, the above theorem allows one to compute the fractal dimension of the attractor of a PAA by converting it into an equivalent recurrent IFS, and then applying the results of [4, 12].

A *weighted* PAA is a PAA in which a weight, between 0 and 1 inclusive, is assigned to each state, and there are no final states. A point generated at a state is chosen to be in the attractor, with a probability equal to the weight of the state. Clearly, every PAA is equivalent to a weighted PAA in which a state has weight 0 (or 1) if it is a nonfinal (or final) state.

We now show that weighted PAA are not more powerful than PAA.

Theorem 3.6. *For every weighted PAA M_1, there exists an equivalent PAA M_2.*

Proof. Construct a PAA M_2 from M_1 as follows. Let a state s in M_1 have weight p. Replace it by two states s_1 and s_2. Mark s_1 as final and s_2 as nonfinal. All incoming transitions of s are the incoming transitions of s_1 and s_2, except for the change in the associated probabilities. If a state q has a transition to state s having probability a then it has transitions to s_1 and s_2 having probabilities $a \cdot p$ and $a \cdot (1-p)$, respectively. If state s has a self-loop having probability a, then s_1 and s_2 have self-loops having probabilities $a \cdot (1-p)$ and $a \cdot p$, respectively, and s_1 has a transition to s_2 with probability $a \cdot p$, and s_2 has a transition to s_1 with probability $a \cdot (1-p)$. All outgoing transitions of s are the outgoing transitions of s_1 and s_2. Clearly, the attractor of M_2 is same as that of M_1. □

4. Affine regular sets

So far we have not shown that the existence and uniqueness of the attractor of a PAA is guaranteed and that the chaos game algorithm on a PAA always (probabilistically) results in an approximation to its attractor. In this and the next section, we will introduce some other generalizations of the IFS method to generate images. This would help us in understanding PAA more clearly by giving us a rigorous characterization of their attractors, as all these generalizations turn out to be equivalent in their capacity to generate images.

Definition 4.9. Let $\{w_1, w_2, \ldots, w_N\}$ be N affine transformations on a compact metric space X. Let $\Sigma = \{1, 2, \ldots, N\}$ represent these N transformations. Σ is called the underlying *code alphabet*. A regular set over Σ is called an *affine regular set*, effectively given as an *affine expression*, which is nothing but a regular expression over Σ. An affine regular set is required to satisfy loop contractivity, that is, it is accepted by a finite automaton satisfying the loop contractivity condition.

Lemma 4.2. *Let Σ be a code alphabet representing N affine transformations. Let $R \subseteq \Sigma^*$ be an affine regular set (satisfying loop contractivity). Let $\sigma = \sigma_1\sigma_2\sigma_3 \ldots \in R$. Then, for all $x \in X$,*

$$\varphi(\sigma) = \lim_{j \to \infty} w_{\sigma_1}(w_{\sigma_2}(\ldots w_{\sigma_j}(x)\ldots))$$

exists and is independent of x.

Proof. Define

$$\phi(\sigma, n, x) = w_{\sigma_1}(w_{\sigma_2}(\ldots w_{\sigma_n}(x)\ldots)).$$

Let M be an ω-FA accepting adherence(R). In the transition diagram of M, let s be the maximum of the contractivities of all loops, let t be the maximum of the contractivities of all transitions, and let l be the maximum of all loop lengths.

Consider the run of M on $\sigma_1\sigma_2\ldots\sigma_n$. During this run, M enters loops at least $\lfloor n/l \rfloor$ times, and then follows a simple path of length at most $k-1$, where k is the number of states in M. Let $x_1, x_2 \in X$ and $m, n \in \mathbf{N}$, and suppose that $m \leqslant n$. Then

$$\phi(\sigma, n, x_2) = \phi(\sigma, m, \phi(\omega, n-m, x_2)),$$

where $\omega = \sigma_{m+1}\sigma_{m+2}\ldots\sigma_n\ldots \in \Sigma^\omega$. Let $x_3 = \phi(\omega, n-m, x_2)$. Therefore,

$$d(\phi(\sigma, m, x_1), \phi(\sigma, n, x_2)) \leqslant s^{\lfloor n/l \rfloor}\, t^{k-1}\, d(x_1, x_3) \leqslant s^{\lfloor n/l \rfloor}\, t^{k-1} D,$$

where $D = \max\{d(x_1, x_3) \mid x_1, x_3 \in X\}$ is a finite constant. Since t and k are constants, we have

$$d(\phi(\sigma, m, x_1), \phi(\sigma, n, x_2)) \leqslant s^{\lfloor n/l \rfloor} D'.$$

Therefore, since $s < 1$, the right-hand side tends to zero as m and n tend to infinity. From the compactness of X, this implies that $\varphi(\sigma)$ exists and is independent of $x \in X$. $\square$

Now we formally state the definition of the image represented by an affine regular set. This definition is justified by the above lemma.

Definition 4.3. Let $R \subseteq \Sigma$ be an affine regular set, where Σ is the code alphabet representing some affine transformations on a compact metric space X. Let $\varphi: \Sigma^\omega \to X$ be the function defined as in Lemma 4.2. Then the image defined by R is φ(adherence(R)), and is called its attractor.

5. Mutually recursive function systems

Barnsley's deterministic algorithm can be described by the recursive formula

$$S^m = w_{i_1}(S^{m-1}) \cup w_{i_2}(S^{m-1}) \cup \cdots \cup w_{i_N}(S^{m-1}), \quad m = 0, 1, 2, \ldots$$

defined on one variable S, which can have a compact subset of the underlying compact space X as its "value." The superscript m indicates the iteration step. S^0 is an arbitrary compact subset of X. We can generalize the algorithm to more than one mutually recursive formulas, each of the form

$$S_i^m = w_{i_1}(S_{j_1}^{m-1}) \cup w_{i_2}(S_{j_2}^{m-1}) \cup \cdots \cup w_{i_r}(S_{j_r}^{m-1}) \tag{1}$$

(where r depends on i) defined on n variables $S_1, S_2, \ldots, S_n$. In other words, each variable is defined in terms of the union of images of some other variables under affine transformations.

Such $n \geqslant 1$ mutually recursive definitions specify a mutually recursive IFS provided they satisfy loop contractivity in the following sense. The interrelationship of the variables can be represented by a digraph with n nodes (or states). Each variable is represented by a state and has incoming transitions from all those states which represent the variables appearing on the right-hand side of the definition of the variable. Let the variables $S_1, S_2, \ldots, S_n$ be represented by states $s_1, s_2, \ldots, s_n$. Therefore, if variable S_i is defined as in the above definition (eq. (1)), then the state s_i has incoming transitions from states $s_{j_1}, s_{j_2}, \ldots, s_{j_r}$ and labeled with affine transformations $w_{i_1}, w_{i_2}, \ldots w_{i_r}$, respectively. A set of mutually recursive definitions is said to satisfy the loop contractivity condition if the digraph representing these definitions satisfies loop contractivity.

Definition 5.1. A *deterministic mutually recursive FS* (DMRFS) is a set of mutually recursive definitions of $n \geqslant 1$ variables, each variable defined as a union of images of some other variables under affine mappings, such that they satisfy the loop contractivity condition. Some variables are marked as "final" or "display" variables.

Note that a DMRFS M need not satisfy the strong connectivity condition.

The attractor of a DMRFS M is computed by selecting n arbitrary nonempty compact sets $S_1^0, S_2^0, \ldots, S_n^0$ as the initial values of the variables and applying the mutually recursive definitions to compute iteratively their new values. The attractor is the union of the limits of the values of final variables

$$A(M) = \lim_{m \to \infty} \bigcup_{S_i \in F} S_i^m,$$

where F is the set of final variables.

Alternatively, $A(M)$ can be expressed in terms of the fixed point of a mapping $W: \mathcal{H}(X^n) \to \mathcal{H}(X^n)$ defined such that

$$W((S_1^{m-1}, S_2^{m-1}, \ldots, S_n^{m-1})) = (S_1^m, S_2^m, \ldots, S_n^m).$$

That is, there exists a unique assignment of values, say $S_1', S_2', \ldots, S_n'$, to variables which remains invariant under one single application of the mutually recursive definitions. Then

$$A(M) = \bigcup_{S_i \in F} S_i'.$$

The existence and uniqueness of such a fixed point follows from the loop contractivity condition [16].

Probabilistic mutually recursive FS. Suppose, with every mutually recursive formula of a DMRFS,

$$S_i^m = w_{i_1}(S_{j_1}^{m-1}) \cup w_{i_2}(S_{j_2}^{m-1}) \cup \cdots \cup w_{i_r}(S_{j_r}^{m-1}),$$

we associate r probabilities $p_{i_1}, p_{i_2}, \ldots, p_{i_r}$, such that $\sum_{k=1}^{r} p_{i_r} = 1$. That is, we specify the weight of the contribution that each of S_{i_k}'s makes in the computtion of S_i. In other words, we assign probabilities to the edges of the digraph representing the DMRFS such that probabilities assigned to *incoming* edges for each state sum to unity. (Note that in the case of PAA, the probabilities on *outgoing* transitions summed to unity.) Such a mutually recursive FS is called a *probabilistic mutually recursive* FS (PMRFS).

The attractor of a PMRFS M can be computed by the following algorithm, which is a generalization of the chaos game algorithm on IFS: Initially, n points $x_1^0, x_2^0, \ldots, x_n^0$ in X are chosen randomly. Consider the point x_i^0 to be associated with the state s_i. At each step of the algorithm, each state chooses one of the incoming transitions according to the assigned probabilities. Let, for $1 \leqslant i \leqslant n$, the state s_i choose an incoming transition from state s_{j_i} labeled with transformation w_{l_i}. Then we have

$$x_i^m = w_{l_i}(x_{j_i}^{m-1}), \quad m = 1, 2, 3, \ldots$$

The collection of points $\bigcup_{s_i \in F} \{x_i^m\}_{m=0}^{\infty}$ defines the attractor $A(M)$ of the PMRFS M.

A probabilistic MRFS defines, in addition to the final attractor, a texture of the image which is determined by the probabilities. The texture is formalized as a probabilistic measure in exactly the same manner as done for PAA in Section 3. Moreover, just like IFS, the class of the supports of attractors defined by probabilistic MRFS is same as that defined by deterministic MRFS.

It is interesting to note that a DMRFS can be implemented by an IFS on a metric space of higher dimension, when the attractor defined by the IFS is *projected* onto the metric space of the DMRFS.

Precisely speaking, let M be a DMRFS with k states and let X be the metric space. Define the projection operator P_k, which maps a set B in X^k to a set in X, as follows:

$$P_k(B) = \{x \mid \exists (x_1, x_2, \ldots, x_k) \in B \text{ such that } x = x_r \text{ for some } r,\ 1 \leqslant r \leqslant k\}.$$

Then there exists an IFS I on metric space X^k such that $A(M) = P_k(A(I))$. For details of this result, see [14], which is informally stated in the following theorem.

Theorem 5.2. *Let M be a DMRFS (or a PMRFS) on n variables, and let X be the underlying compact metric space X. Then there exists a higher-dimensional IFS whose projected attractor is same as that of M.*

Proof. Consider the digraph representing M. Let d_i be the incoming degree of state s_i, $1 \leqslant i \leqslant n$. Then define an IFS I on metric space X^n with $d_1 \times d_2 \times \cdots \times d_n$ transformations. Each transformation represents some possible combination by which each variable is computed from one of its neighboring variables connected by an incoming transition. Then the deterministic algorithm on I defines the attractor in exactly the same way as defined by M viewed as a DMRFS. Furthermore, the chaos game algorithm on I defines the attractor in exactly the same way as defined by M viewed as a PMRFS. $\square$

The above theorem also shows the equivalence of DMRFS and PMRFS in terms of their power to generate images (as compact sets). Therefore, in this context we will refer to them jointly as MRFS.

5.1. Equivalence of MFRS and affine regular sets

Theorem 5.3. MRFS *and affine regular sets define the same class of images (as compact sets).*

Proof. Let M be an MRFS. Construct an FA M' from M as follows. Reverse the directions of the edges of the digraph representing M (simultaneously changing the labeling transformations to the corresponding code symbols). Create a new initial state which has ε-transitions to every final state of M. Note that all the states of M' can be marked as final as the adherence of the affine regular set defines an image. Now we claim that $a \in A(M)$ iff $a \in \varphi(\text{adherence}(L(M')))$.

Let $a \in A(M)$. Let $S'_1, S'_2, \ldots, S'_n$ be the fixed point of the mapping $W: \mathscr{K}(X^n) \to \mathscr{K}(X^n)$ defined by M. Then $a \in S'_i$ for some final variable S_i. Let S_i be defined as

$$S_i = w_{i_1}(S_{j_1}) \cup w_{i_2}(S_{j_2}) \cup \cdots \cup w_{i_r}(S_{j_r}).$$

Therefore, $a \in w_{i_k}(S_{j_k})$ for some $k \in \{1, 2, \ldots, r\}$, i.e. $a = w_{i_k}(x_1)$ for some $x_1 \in S'_{j_k}$. We can apply the above to x_1, and obtain sequences $\sigma_1, \sigma_2, \sigma_3, \ldots$ and $x_1, x_2, x_3, \ldots$ such that

$$a = w_{\sigma_1}(w_{\sigma_2}(\ldots w_{\sigma_j}(x_j)\ldots)).$$

Note that $w_{\sigma_1} w_{\sigma_2} \ldots w_{\sigma_j}$ is the labeling of a path of length j in M in *reverse* direction, starting from a final variable to which a belong to. This implies that $\sigma_1 \sigma_2 \ldots \sigma_j$ is the labeling of a corresponding path in M', starting at the initial state. Furthermore, for any $x \in X$, we have

$$\begin{aligned} &d(a, w_{\sigma_1}(w_{\sigma_2}(\ldots w_{\sigma_j}(x)\ldots))) \\ &\quad = d(w_{\sigma_1}(w_{\sigma_2}(\ldots w_{\sigma_j}(x)\ldots)), w_{\sigma_1}(w_{\sigma_2}(\ldots w_{\sigma_j}(x_j)\ldots))) \\ &\quad < s^{\lfloor j/l \rfloor} t^{l-1} d(x_j, x) < s^{\lfloor j/l \rfloor} D, \end{aligned}$$

where l is the maximum of the loop lengths, s is the maximum of the loop contractivities, and t is the maximum of all edge contractivities in the transition diagram of M. As

$j \to \infty$, the right-hand side tends to zero. From the compactness of X, we conclude that

$$a = \lim_{j \to \infty} w_{\sigma_1}(w_{\sigma_2}(\ldots w_{\sigma_j}(x)\ldots)) \tag{2}$$

for all $x \in X$, i.e. $a = \varphi(\sigma_1 \sigma_2 \sigma_3 \ldots)$, where $\sigma \in \text{adherence}(L(M'))$.

Conversely, suppose (2) holds. Then, choosing x to be in the fixed point $\bigcup_{i=1}^{n} S'_i$, where $(S'_1, S'_2, \ldots, S'_n)$ is the fixed point of M, we conclude that a is in the attractor of M.

Now, let R be an affine expression accepted by an FA M. Construct an MRFS M' from M as follows. Reverse the directions of M, make the initial state of M the only final state of M'. Then, from exactly the same argument as above, the equivalence of M and M' can be concluded. □

Corollary 5.4. *Let* $\{X; w_1, w_2, \ldots, w_N\}$ *be an* IFS. *Then* Σ^*, *where* Σ *is the corresponding code alphabet, is an equivalent affine expression.*

6. Equivalence of PAA and affine regular sets

Now we are in a position to characterize the attractor of a PAA. However, we are not yet ready to show the equivalence of PAA and MRFS, as the latter need not satisfy the strong connectivity condition. We now show that MRFS can be restricted to always satisfy this condition.

Theorem 6.1. *Let M be an* MRFS. *Then there exists a strongly connected* MRFS *M' which defines the same image (as compact set) as the one defined by M.*

Proof. Let M be an MRFS defined on n variables $S_1, S_2, \ldots, S_n$. Create an $(n+1)$th variable S_{n+1}. Let $(S'_1, S'_2, \ldots, S'_n)$ denote the fixed point of M. Now, without loss of generality, we can assume that each of S'_i is nonempty. Let $x_i \in S'_i$, $1 \leqslant i \leqslant n$. Define n transformations u_i, $1 \leqslant i \leqslant n$, by $u_i(x) = x_i$, for all $x \in X$. Now, if S_i is defined as

$$S_i^m = w_{i_1}(S_{j_1}^{m-1}) \cup w_{i_2}(S_{j_2}^{m-1}) \cup \cdots \cup w_{i_r}(S_{j_r}^{m-1}),$$

then change this definition to

$$S_i^m = w_{i_1}(S_{j_1}^{m-1}) \cup w_{i_2}(S_{j_2}^{m-1}) \cup \cdots \cup w_{i_r}(S_{j_r}^{m-1}) \cup u_i(S_{n+1}^{m-1}).$$

In other words, S_{n+1} contributes x_i to the computation of S_i. Also, each S_i contributes x_i to S_{n+1}, i.e.

$$S_{n+1}^m = u_1(S_1^{m-1}) \cup u_2(S_2^{m-1}) \cup \cdots \cup u_n(S_n^{m-1}).$$

S_{n+1} is marked as nonfinal. It is easy to verify that this addition of new variable does not affect the attractor, as no new points are ever computed. Moreover, there is a path from every variable to every other variable via S_{n+1}, and, therefore, the resulting MRFS is strongly connected.

We need the following two lemmas.

Lemma 6.2. *Let M be a* PAA *and let $u = w_1 w_2 \ldots w_n$ be the labeling of a finite path in the transition diagram of M. Let the chaos game algorithm traverse an ω-path $\tau = w_{i_1} w_{i_2} w_{i_3} \ldots$ Then, with probability almost one, u is a subword of τ.*

The above lemma follows from the elementary probability theory.

Lemma 6.3. *Let R be an affine regular set and let $\sigma \in \text{adherence}(R)$. Let $a = \varphi(\sigma)$, i.e.*

$$a = \lim_{j \to \infty} w_{\sigma_1}(w_{\sigma_2}(\ldots w_{\sigma_j}(x)\ldots))$$

for all $x \in X$. Let $\delta > 0$. Then there exists m such that

$$w_{\sigma_1}(w_{\sigma_2}(\ldots w_{\sigma_m}(x)\ldots))$$

is in δ-neighborhood of a, for all $x \in X$.

Proof. As shown in the proof of Theorem 5.3,

$$d(a, w_{\sigma_1}(w_{\sigma_2}(\ldots w_{\sigma_m}(x)\ldots))) < s^{\lfloor m/l \rfloor} D,$$

where s, l and D are positive constants. Choose m large enough so that $s^{\lfloor m/l \rfloor} D \leqslant \delta$. □

The above lemma says that we can visit smaller and smaller neighborhoods of a point a in the attractor by applying longer and longer prefixes of its "address" σ to some arbitrarily chosen points in the underlying metric space X.

Theorem 6.4. PAA *and affine regular sets define the same class of images.*

Proof. Let R be an affine regular set accepted by an FA M. Convert M into a strongly connected FA M' which accepts an affine regular set R' such that R and R' define the same image (apply Theorems 5.3 and 6.1). Now convert M' into a PAA M'' as follows: reverse the directions of transitions of M' and make its initial state the only final state of M''. Let A denote the set Prefix (adherence(R')). Let the chaos game algorithm on M'' traverse the ω-path labeled by $w_{\sigma_1} w_{\sigma_2} w_{\sigma_3} \ldots$ Let $\sigma = \sigma_1 \sigma_2 \sigma_3 \ldots$

Consider all those subwords of σ which were traversed during the execution of the chaos game algorithm such that, after this traversal, the finite control of M'' was in the final state. Let B be the set of all such finite subwords of σ. Then, from Lemma 6.2, any word in Reversal(B) is in A and vice versa, with probability almost one.

In other words, during the execution of the chaos game algorithm, we traverse longer and longer finite paths in M'', which means that we are approximating points by applying longer and longer prefixes of their addresses, i.e. ω-words in adherence(R'). By Lemma 6.3, we are visiting smaller and smaller neighborhoods of points precisely in φ(adherence(R')). Hence, M'' generates the same image as R.

For the converse, let M be a PAA. Convert it into an FA M' by reversing the directions of transitions of M, creating a new initial state which has ε-transitions to the final states of M. Then, from the same argument as above, we conclude that $L(M')$ defines the same image as generated by M. $\square$

Considering images as measures (greyness, color), it seems difficult to show that for every PAA there exists an equivalent PMRFS or vice versa. However, it follows from Theorems 5.3 and 6.4 that we can convert every PAA into a PMRFS that generates a measure with the same support, or vice versa. For strongly connected PMRFS, we can show this directly as follows.

Consider the chaos game algorithm on a PAA M, first when it is viewed as a PAA and, second, when it is viewed as a PMRFS (all nonzero probabilities are, however, changed to some nonzero values such that they satisfy the requirement that incoming probabilities sum to unity). In both cases, the set of the finite sequences of affine transformations applied to some points in the underlying compact metric space is (probabilistically) same and, therefore, by Lemma 6.3, we are approximating the same attractor in both cases. The only difference is that, in the first case, we generate these finite sequences as subwords of one infinite sequence, whereas, in the latter case, we generate them in a parallel tree-like fashion.

Therefore, the equivalence of PAA with affine expressions and MRFS provides us with the explanation why a PAA always defines uniquely an image and why the chaos game algorithm always approximates it.

6.1. Problem of image encoding by MRFS (or PAA)

For IFS, Barnsley's collage theorem provides the mathematical basis for automatically inferring the IFS "code" of a given image [1]. It states that, given an image B in $\mathscr{H}(X)$, if an IFS with the contraction mapping W is chosen and, for some $\varepsilon>0$,

$$h(B, W(B))\leqslant\varepsilon,$$

then

$$h(B, A)\leqslant\frac{\varepsilon}{1-s},$$

where A is the attractor of the IFS and s is the contractivity factor of W.

The theorem can be generalized to the case when we want to infer an MRFS (or a PAA) from a given image, in a way analogous to the one by which it was generalized to recurrent IFS in [2]. This provides the basis for an interactive trial-and-error method of coming up with an MRFS for a given image on a computer screen, in which the user "guesses" an MRFS, whose one iterative step when applied to the image results in an image which is quite "close" to the original image. The collage theorem then guarantees that the attractor of the MRFS will be also quite "close" to the given image.

7. Applications

In this section, we give some examples illustrating how PAA and MRFS can be employed to define images and also how they can implement some other known methods to generate images and fractals.

7.1. Image generation and compression

Affine automata and MRFS have clear applications in image generation and compression. Images can be generated both as black points (as compact sets) or as grey tones or color tones (as measures).

Example. As an example illustrating the power of PAA to describe natural objects, see Fig. 3d for the generation of a fern. The fern is generated by the PAA shown in Fig. 3a, in which both the states are final states. The numbers in parenthesis are the probabilities. The four affine transformations are the same as given in the last example in Section 3 on IFS, which generated the fern shown in Fig. 1b. Denote the set of points generated at states s_1 and s_2 by S_1 and S_2, respectively. The sets S_1 and S_2 are shown in Fig. 3b and 3c, respectively. The attractor is the union of these two sets. Note that the branches of this fern are one-sided. This fern cannot be generated by an IFS. It is possible to generate a "self-similar" fern by an IFS, which has two-sided branches, as shown in Fig. 1b.

The "one-sided" fern is also defined by an affine expression over code alphabet $\Sigma=\{1,2,3,4\}$, representing the four affine transformations defined above. The affine expression is

$$((1+2)^*3(1+2)^*4)^*+((1+2)^*4(1+2)^*3)^*.$$

Alternatively, this fern is the attractor of the DMRFS M:

$$S_1^m=w_1(S_1^{m-1})\cup w_2(S_1^{m-1})\cup w_4(S_2^{m-1}),$$

$$S_2^m=w_3(S_1^{m-1})\cup w_1(S_2^{m-1})\cup w_2(S_2^{m-1}).$$

Let (S_1, S_2) be the fixed point (attractor) of M. The sets S_1 and S_2 are shown in Fig. 3b and 3c, respectively. The attractor is the union of these two sets. Finally, note that the diagram in Fig. 3a also represents a PMRFS generating the same image, as the requirement that incoming probabilities should sum to unity is satisfied.

Example. For another example, see Fig. 3e, which shows a complex "recursive" image of a tree along with its "shadow." There is a mirror hanging from one of the branches which contains infinitely deep images of the whole image. This image is generated by

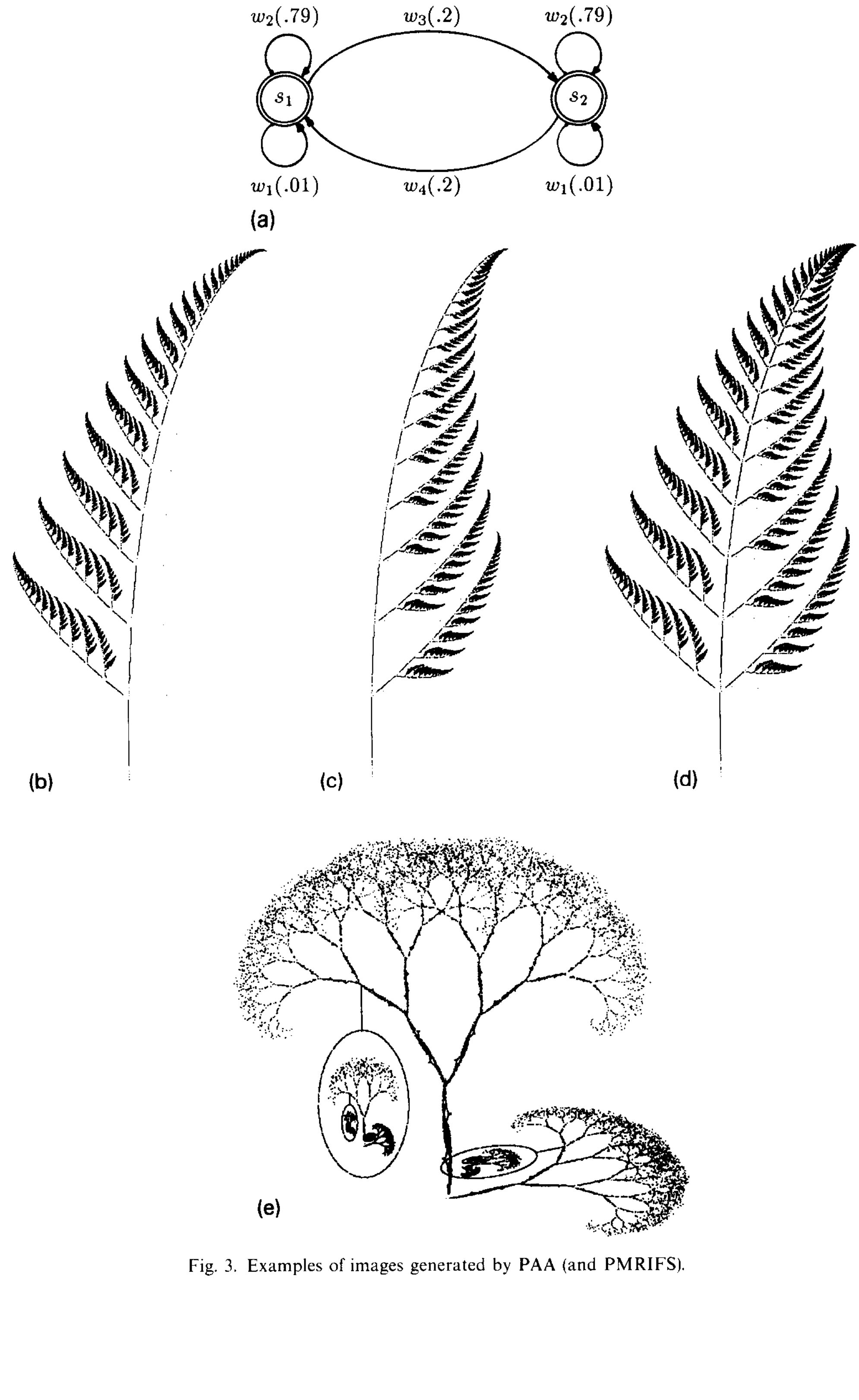


Fig. 3. Examples of images generated by PAA (and PMRIFS).

a six-state PAA. To give the reader an idea of how this image is generated, we mention that if a part of an image is generated at a state, say s, then we can create a "shadow" of this part by transiting to a state, say r, where the transition is labeled with an appropriate affine transformation w, which rotates and scales down the part of the image created at the state s. As a programming trick, we then go back to the state s from the state r, by a transition labeled with transformation w^{-1}.

Example. One can even combine the chaos game algorithm and the deterministic algorithm in the following interesting way: generate, let us say, 2000 points approximating the attractor of an MRFS by the chaos game algorithm and then run the deterministic algorithm with these 2000 points as the input, i.e. the initial "values" of the variables. Furthermore, more than one MRFS can be interconnected in a similar manner, with the outputs of one MRFS serving as the inputs of some other MRFS. Some images generated by such a *hybrid algorithm* are shown in Fig. 4, in which we have used the MRFS generating the image in Fig. 3 to obtain a more complex sequence of such images. Note that the mirror hanging from the branch of a tree in this sequence now contains an image of the remaining sequence. In Fig. 4a and 4b, the tree, which is the output of the first MRFS, is generated by the probabilistic and

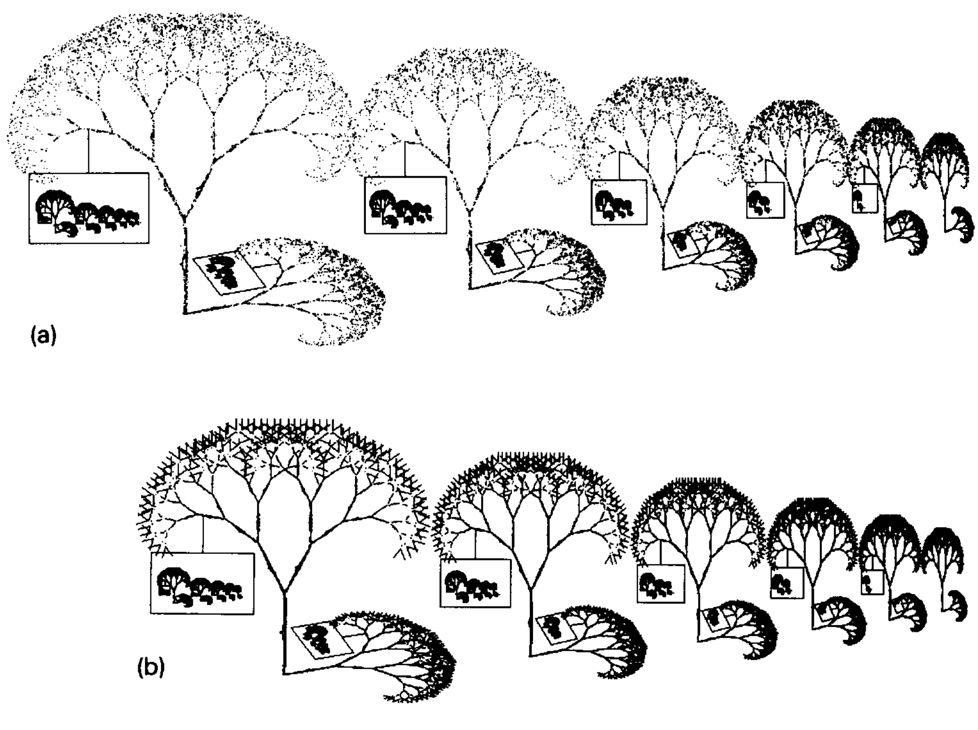


Fig. 4. Examples of images generated by the hybrid algorithm.

deterministic algorithms, respectively. This becomes the input of another MRFS, which then generates the sequence of trees.

Sometimes, texture is defined as a spatial arrangement of some basic primitive elements and is an indication of the "graininess" or "coarseness" of the image. Such texture images can be generated very elegantly by deterministic MRFS. In the generation of images with some "coarse" texture we are not interested in the final attractor of the MRFS but rather in the output of the deterministic algorithm after some finite number of iterations, when the desired graininess is obtained.

7.2. *Implementing rational expressions by PAA and MRFS*

Some automata-theoretic methods to generate images have recently been studied. In [9], adherences of rational sets have been studied as a tool for image compression. Some interesting patterns have been shown to be represented compactly by finite automata [5].

PAA (and, therefore, MRFS) constitute a strict generalization of these automata-theoretic techniques to define images and, therefore, provide us with an efficient mechanism to implement these techniques on standard hardware that is fast for arithmetic computations.

We now show how a rational expression, used in [9] as a tool to define images, can be efficiently implemented by a PAA (or an MRFS). Considering binary notation, a string over alphabet $\Sigma=\{(0,0),(0,1),(1,0),(1,1)\}$, say $(0,1)(1,1)(1,0)$, is interpreted as a 2-D point $(0.011, 0.110)$. Correspondingly, an ω-word is interpreted as a 2-D point with real coordinates. Finally, the 2-D image represented by a regular set $\rho\subseteq\Sigma^*$ is the set of 2-D points obtained by interpreting adherence(R). See [9] for more details.

Theorem 7.1. *The class of images defined by rational expressions is a proper subset of the one defined by affine expressions.*

Proof (*outline*). We will prove the theorem for 2-D images defined by rational expressions over binary alphabet. Let R be a rational expression over alphabet $\Sigma=\{0,1\}$ representing a 2-D image. Consider the four contractive transformations which map the unit square onto one of the quadrants:

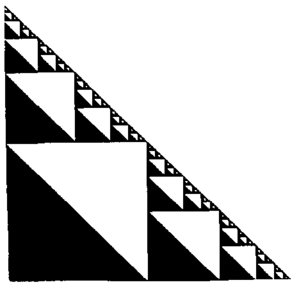
$$w_1(x,y)=(0.5x,0.5y),\qquad w_2(x,y)=(0.5x,0.5y+0.5),$$

$$w_3(x,y)=(0.5x+0.5,0.5y),\qquad w_4(x,y)=(0.5x+0.5,0.5y+0.5).$$

Let $\Delta=\{1,2,3,4\}$ be the underlying code alphabet. Consider the morphism $h:\Sigma\times\Sigma\to\Delta$ defined as

$$h(0,0)=1,\qquad h(0,1)=2,\qquad h(1,0)=3\quad\text{and}\quad h(1,1)=4.$$

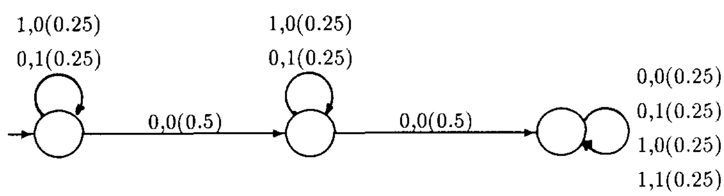
Now $h(R)\subseteq\Delta^*$ is an affine expression which defines the same image as the one defined by the rational expression R. For details, see [9]. $\square$



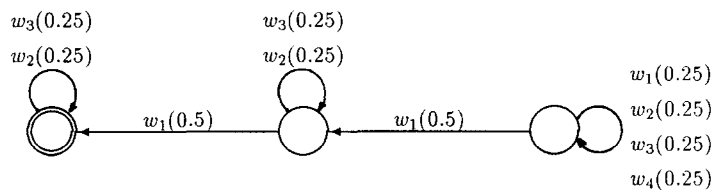
(a) Image

$$I = \{(0,1) + (1,0)\}$$
$$\Delta = \{I + (0,0) + (1,1)\}$$
$$I^*(0,0)I^*(0,0)\Delta^*$$

(b) Rational Expression



(c) Probabilistic Finite Generator



(d) Equivalent PMRFS

Fig. 5. Implementing a rational expression by a PMRFS.

We can use the above theorem and Theorems 5.3 and 6.4 to implement efficiently rational expressions by MRFS and PAA. As an example, see Fig. 5a, which shows an image represented by a rational expression. The finite generator (automaton) accepting the rational expression is also shown. This image can be generated by a PMRFS, which is shown in Fig. 5d. The labeling transformations w_i's are the ones defined in the proof of the above theorem.

8. Conclusions

We consider various generalizations of the IFS method to generate images. The probabilistic affine automata constitute a powerful mechanism to generate highly complex images. They are equivalent to affine regular sets and mutually recursive IFS.

Clearly, PAA and MRFS are capable of compressing drastically the description of images and possibly other forms of data. Self-similar fractals like the Sierpinski triangle, ferns, clouds, etc., happen to be only a subset of the images generated by these techniques. As an avenue of future research, the design of efficient practical algorithms to infer a PAA or an MRFS from a given image is a challenging problem, with great potential value in image compression.

Note added in proof

Recently, various extensions of IFS have been proposed. It seems that the MRFS introduced here and in particular the MRFS with control strings considered in [18, 19] are the most powerful and efficient tools for image generation.

References

[1] M.F. Barnsley, *Fractals Everywhere* (Academic Press, New York, 1988).

[2] M.F. Barnsley, J.H. Elton and D.P. Hardin, Recurrent iterated function systems, *Constr. Approx.* **5** (1989) 3–31.

[3] M.F. Barnsley, R.L. Devaney, B.B. Mandelbrot, H-O. Peitgen, De Saupe and R.F. Voss, *Science of Fractal Images* (Springer, Berlin, 1988).

[4] M.F. Barnsley, A. Jacquin, L. Reuter and A.D. Sloan, Harnessing chaos for image synthesis, in: *Computer Graphics, Proc. SIGGARPH' 1988 Conf.*

[5] J. Berstel and M. Morcrette, Compact representation of patterns by finite automata, in: *Proc. Pixim '89*, Paris, pp. 387–402.

[6] L. Boasson and M. Nivat, Adherences of languages, *J. Comput. System Sci.* **20** (1980) 285–309.

[7] R. Cohen and A. Gold, Theory of ω-languages, Part I and II, *J. Comput. System Sci.* **15** (1977) 169–208.

[8] K. Culik and S. Dube, L-systems and MRFS, manuscript, 1991.

[9] K. Culik and S. Dube, Rational and affine expressions for image description, Tech. Report TR90001, Dept. of Computer Science, Univ. of S. Carolina; *Discrete Appl. Math.*, to appear.

[10] K. Culik and S. Yu, Cellular automata, $\omega\omega$-regular sets, and sofic systems, *Discrete Appl. Math.*, to appear.

[11] F.M. Dekking, Recurrent sets, *Adv. in Math.* **44** (1982) 78–104.

[12] D.B. Ellis and M.G. Branton, Non-self-similar attractors of hyberbolic IFS, in: J.C. Alexander, ed., *Dynamical Systems*, Lecture Notes in Mathematics, Vol. 1342 (Springer, Berlin, 1988) 158–171.

[13] J. Gleick, *Chaos – Making a New Science* (Penguin, Baltimore, MD, 1988).

[14] Y. Liu, Recurrent IFS, ω-orbit finite automata, and regular set plotter, M.S. Thesis, Dept. of Comp. Sci., Univ. of S. Carolina, 1990.

[15] B. Mandelbrot, *The Fractal Geometry of Nature* (Freeman, San Francisco, 1982).

[16] R.D. Mauldin and S.C. Williams, Hausdorff dimension in graph directed constructions, *Trans. Am. Math. Soc.* **309** (1988) 811–829.

[17] P. Prusinkiewicz, Applications of L-systems to computer imagery, in: H. Ehrig et al., eds., *Graph Grammars and their Application to Computer Science*, Lecture Notes in Computer Science, Vol. 291 (Springer, Berlin, 1987) 534–548.

[18] K. Culik II and S. Dube, Balancing order and chaos in image generation, *Computer and Graphics*, to appear.

[19] K. Culik II and S. Dube, Balancing order and chaos in image generation, in: *Proc. 18th Internat. Colloq. on Automata, Languages and Programming*, Madrid, Spain, July 1991, Lecture Notes in Computer Science, Vol. 510 (Springer, Berlin, 1991) 600–614.